

REMARKS

Status of this application

In the outstanding Office Action mailed on May 30, 2006, claims 1-4, 6, and 9 were rejected under 35 U.S.C. 103(a) as being unpatentable over Krupa, U.S. Patent Application Publication No. 2002/10156811 (hereinafter "Krupa") in view of the newly cited Jammes et al., U.S. Patent Application Publication No. 2003/0167213 (hereinafter "Jammes"). Claims 5, 7, and 10 was rejected under 35 U.S.C. 103(a) as being unpatentable over Krupa and Jammes in further view of Lee et al. U.S. Patent Application Publication No. 2002/0169788 (hereinafter "Lee"). Claim 8 was rejected under 35 U.S.C. 103(a) as being unpatentable over Krupa in view of Jammes in further view of Harless U.S. Patent Application Publication No. 2003/0005410 (hereinafter "Harless"). Claims 11, 12, and 13 were rejected under 35 U.S.C. 103(a) as being unpatentable over Krupa in view of Jammes, Lee and Harless.

This response amends claim 1 to correct an improper antecedent reference to "said database" to instead refer to "said relational database system." Claim 2 has been canceled and its limitations have been incorporated into claim 3.

This response requests reconsideration of the obviousness rejection of claims 1 and 3-13 for the reasons presented below.

The obviousness rejection based on Krupa in view of Jammes

In rejecting the pending claims, the Examiner has suggested that it would have been obvious to employ the Jammes teaching to modify the system described by Krupa that so that the Krupa system would store an "XML skeleton" document in order to store the hierarchical structure in a source XML document. Reconsideration is requested.

As the Examiner has noted, Krupa teaches a method for storing an XML document in a relational database system in which the XML document is first parsed by an in-memory DOM or JDOM API (see [0025] and [0048]) to identify characters representing data values in the XML document. Krupa then stores these data values in the a specified column location in one or more rows of one or more specified tables (see Krupa Fig. 2 and [0026]).

Thus, although Krupa obtains data characters from the XML document and stores them in relational tables, the Examiner agrees that Krupa does not perform the step recited in claim 1 of *"removing at least some of said characters representing data values from said XML document and storing the remainder of said XML document in said database as an XML skeleton which defines the structure of said XML document and which contains the same characters as the XML document but with said characters representing data values omitted"*.

Moreover, and contrary to the Examiner's suggestion, Krupa does not perform the step of *"thereafter reconstructing said XML document by merging the data content of said specified rows with said XML skeleton."* The Examiner cites the passage in Krupa at [0113] which teaches that an in-memory XML tree can be easily reconstructed by interrogating each data row and creating the appropriate object that corresponds to that row. But that method of reconstructing an XML document does not involve merging the data content from the relational tables with an XML skeleton as claimed. As noted next, Krupa does not create or store an XML skeleton of the type claimed, but instead stores data indicating the hierarchical parent child relationships in the relational tables as seen in Fig. 2 and described at [0014]-[0015] and in more detail at [0036]-[0040].

The Examiner notes that Krupa, at [0024] states that an in-memory XML document tree may be stored in memory with all parent-child relationships and as well as empty elements intact, but an in memory tree (such as a DOM Tree or a JDOM tree) is not an "XML skeleton" as claimed, and the DOM or JDOM in-memory tree is not stored in the relational database system as claimed.

The Examiner agrees that Krupa but does not explicitly teach "storing the remainder of said XML document in said database as an XML skeleton which defines the structure of said XML document and contains the same characters as the XML document but with said characters representing data values omitted" as stated in claim 1, but contends that it would have been obvious to modify the Krupa teaching to incorporate that technique in view of the teaching of Jammes. The Examiner states:

"Jammes teaches storing markup language template files (i.e., HTML) wherein the template files define the structure of the markup document but with data values omitted. Information is extracted on-demand from the database and merged with the markup language template files to construct the markup document (see Abstract and

paragraphs [0012] and [0061-0071]).

Since Krupa and Jammes are both from the same field of endeavor (i.e., both markup languages are descendants of SGML), the motivational purpose of a more efficient means of storing and retrieving markup documents via a database as disclosed by Jammes would have been recognized in the pertinent art of Krupa. It would have been obvious at the time the invention was made to a person having ordinary skill in the art to modify the teaching of Krupa with the teachings of Jammes."

First, as the Examiner recognizes, the HTML template files that Jammes employs are not XML documents but rather template web pages. Jammes template web pages are not formed by "removing at least some of the character representing data values from XML document and storing the remainder of said XML document" but instead are created using a HTML authoring tool as described at [0062]. Furthermore, Jammes template web pages are not stored in a relational database required by claim 1 but are rather stored as separate files on accessible storage media as Jammes states at [0061]. Moreover, when the Jammes system merges data from the database with a web page template, it effectively performs a "fill in the blanks" function to render the database data in a format more easily understood by the user, but this use of template web pages in no sense performs the claimed step of "reconstructing an XML document" that existed before and from which both data and an XML skeleton were derived and stored in the relational database system as claimed.

It is submitted that Krupa and Jammes have wholly different objectives. Like applicants, Krupa's goal is the storage of XML data in a relational database whereas Jammes seeks to deliver product data to online customers and, to that end, adds product data from a relational database to previously authored web page templates for presentation to customers. There is nothing in either reference that would suggest that the Jammes template web pages might be used to advantage to modify the Krupa method for storing XML documents in a relational database.

Moreover, even if the methods described in Jammes of using template web pages to present relational data to online customers were somehow combined with Krupa's methods, the result would not be the same as the method defined by applicant's claims. Neither Krupa nor James describes removing data characters from an XML document to form an XML skeleton, storing both the removed data and the XML skeleton a relational database system, and thereafter reconstructing the original XML document by merging the data from the relational database with

the XML skeleton as expressly claimed in claim 1.

Claim 1 and its dependent claims 2-13 are accordingly believed to be allowable for the reasons presented above.

Dependent claims 3-13

Dependent claim 2 has been canceled by this response and its limitations have been incorporated into dependent claim 3, upon which the remaining claims 4-13 depend.

Claim 3 as now presented specifies how the data in the XML document is mapped into columns and rows of the relational database tables. Each column contains data from a leaf element that has no subelements, whereas when an element has subelements, the data from each subelement is placed in a separate column of a row which contains the data from the element having subelements.

Krupa does not organize the data from the XML document in this way. As seen in Krupa's Fig. 2, each data value from the XML document goes in the "VALUE" column of a separate row, while the other columns of that row contain information about the data, including its data name, its data type, and data about its parent child relationships. Krupa does not store data from different subelements in an element having subelements in different columns of the same row as required by claim 3 and its dependent claims 4-13.

The Examiner rejected claim 3 stating that the subject matter claimed was described by Krupa - Fig. 2, at [0037]-[0044] and at [0047] *et seq.* in the section describing "The Storage Algorithm". Fig. 2 and the other cited passages in fact make it clear that Krupa places the data values from the XML document only in the VALUE column as described at [0043], and further at [0047] *et seq.* where the data values from the XML document are describes as always being placed in the single column assigned the symbolic name DATA_VALUE.

Accordingly, dependent claims 3 as now presented, as well as the remaining claims 4-13 which are dependent thereon, are believed to be allowable for this additional reason.

The remaining dependent claims are allowable because they incorporate the limitations of parent claims 1 and 3, both of which define important differences over the cited art. In the interest of completeness, applicant makes the following additional brief comments with respect to

selected ones of remaining dependent claims which contain still further limitations clearly not taught by the cited art.

Claim 8 defines a technique for according special treatment to elements designated as "static elements" from which data is not removed when the XML skeleton is formed. The Examiner cites Harless which describes how to parse XML into "static" data structures of the type that can be readily processed by programs written in COBOL. The Harless XML parser does nothing like the function recited in claim 8, so that even if the Krupa method were modified to use the Harless parser, the result would not perform the function recited in claim 8.

Claim 9 describes processing the relational tables produced by the method of claim 1 before that data is merged into the XML skeleton. As the Examiner points out, it is well known that the data in relational tables can be manipulated, but that does not suggest that it would be obvious to produce modified XML documents by performing the combination of steps set forth in claim 9, which is not taught by Krupa. Reconsideration is requested.

Claim 11 and its dependent claims 12-13 describe a mechanism by which data values which are primary key values are not removed but instead retained when the XML skeleton is formed. The Examiner's rejection based on the four way combination of Krupa, James, Lee and Harless is not understood. The Examiner states that Harless designates XML elements as static elements (Harless parses all XML data into fixed length "static" fields records for COBOL processing) but that has nothing to do with the subject matter of claim 11), but none of the cited references even suggests forming an XML skeleton and certainly none suggests that, in the course of doing that, data that is designated as primary key data values should be retained in the skeleton while other data is removed. Reconsideration is requested.

Conclusion

Claim 1 as amended, upon which all of the remaining claims depend, is allowable since it defines method steps which are not performed by the cited Krupa and Jammes systems, even if those two systems were somehow combined, and the proposed combination itself is nowhere disclosed or suggested and would not provide a useful result.

Claim 3 as amended, upon which claims 4-13 depend, defines the manner in which the data in XML documents is mapped into particular columns and rows of relational tables using method steps that are not disclosed by the references. Thus, claims 3 to 13 are allowable for this additional reason.

Claims 8, 9 and 11-13 define additional subject matter that is nowhere disclosed by the cited prior art.

For these reasons, it is believed that claims 1 and 3-12 as now presented clearly define patentable subject matter. This application is accordingly believed to be in condition for allowance, which is requested.

Respectfully submitted,



Dated: August 30 2006

Charles G. Call, Reg. 20,406

Certificate of Transmission under 37 CFR 1.8

I hereby certify that this *Amendment* is being transmitted by facsimile to the central facsimile number of the U.S. Patent and Trademark Office, (571) 273-8300, on August 30, 2006.



Dated: August 30, 2006

Signature _____

Charles G. Call, Reg. No. 20,406
USPTO Customer No. 021253
68 Horse Pond Road
West Yarmouth, MA 02673
Ph. (508) 778-2630 Fax (508) 629-6540
call@patentsoft.com